AMENDMENTS TO THE SPECIFICATION:

Please replace the paragraphs on page 30, lines 12-29 with the following amended paragraphs:

For example, some arrangements use a trusted element to perform certain secure functions, but perform other tasks within an insecure environment. Figure 15 shows an example electronic appliance 61 including a trusted element 109 such as a secure execution space ~~108~~ and an insecure execution space 550. Appliance 61 might, for example, be a personal computer providing trusted element ~~108~~ 109 in the form of a hardware or software tamper-resistant protected processing environment; and insecure execution space 550 in the form of the personal computer processor's typical execution space. It may be desirable to permit an application (e.g., a program) 600 executing within insecure execution space 550 to request services from trusted element ~~108~~ 109. In this scenario, it would be desirable to allow a validation authority 100 to certify the application (e.g., to ensure that the application follows rules for good application behavior) – and allow the trusted element ~~108~~ 109 to validate the application's certification before providing any services to it. For example, trusted element ~~108~~ 109 can refuse to provide a requested service if application 600 has not been certified or if application 600 has been tampered with.

Since insecure execution space 550 does not provide the tamper-resistance necessary to support truly secure validation of application 600, it would be desirable to provide a tamper-resistant mechanism for allowing trusted element ~~108~~ 109 to validate certifications presented by applications intended to be run or otherwise used, at least in part, within an insecure environment.

Please replace the paragraph on page 31, lines 13-22 with the following amended paragraph:

To validate the credential, the trusted element ~~108~~ 109 may authenticate the credential, and then issue challenges based on different parts of the authenticated credential that the trusted element selects in an unpredictable (e.g., random) way. For example, the trusted element ~~108~~ 109 can repeatedly challenge application 600 or other agent to provide (or it can itself generate) a cryptographic hash value corresponding to application portions the trusted element ~~108~~ 109 randomly selects. The trusted element ~~108~~ 109 can compare the responses to its challenges with information the authenticated credential provides, and deny service to application 600 or take other appropriate action if the comparison fails. The challenges may be repeated on an ongoing basis (e.g., during execution of application 600) and/or interleaved with non-predetermined challenges not defined by the credential, to increase the tamper-resistance of the verification process.

Please replace the paragraph on page 34, line 26 through page 35, line 2, with the following amended paragraph:

Encryption key 762 may be chosen from a set of global credential encryption keys, such that the corresponding decryption key is guaranteed to be present at the trusted element ~~108~~ 109 where the application 600 will be used. Different encryption keys 762 can be used to distinguish among applications suitable for different environments, as described above. The signature process 751 and encryption process 752 may be applied in any order and to any arbitrary selection or

subsets 745 of hash blocks 740 such that the result protects the contents of hash blocks from disclosure and/or modification.

Please replace the paragraphs on page 35, lines 10-21, with the following amended paragraphs:

Figure 21 shows an overall example credential validation process 900 performed by appliance 61. In this example, appliance 61 includes a trusted element ~~108~~ 109 (e.g., a protected processing environment) providing a validator function 920. In this example, validation process 900 takes information from distribution medium 616 (which may have been copied to other media) and presents it to appliance 61 for validation by validator 920 within trusted element ~~108~~ 109. Thus, in this example, it is trusted element ~~108~~ 109, not application 600, that is trusted – and the trusted element is responsible for validating the application before the trusted element will provide any services to the application.

In this particular example, when appliance 61 begins to use or execute application 600, trusted element ~~108~~ 109 performs a validation process in which credential 612 is presented to validator 920 along with data calculated by a "select" process based on application 600. The validator 920 determines whether credential 612 is a valid representation of application 600.